ARL-TR-8281 ● JAN 2018

US Army Research Laboratory

Designing for Compressive Sensing: Compressive Art, Camouflage, Fonts, and Quick Response Codes

by Michael L Don

NOTICES

Disclaimers

ARL-TR-8281 ● JAN 2018

US Army Research Laboratory

Designing for Compressive Sensing: Compressive Art, Camouflage, Fonts, and Quick Response Codes

by Michael L Don
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
January 2018	Technical Report	November 2016 – March 2017

4. TITLE AND SUBTITLE
Designing for Compressive Sensing: Compressive Art, Camouflage, Fonts, and Quick Response Codes

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Michael L Don

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-WML-F
Aberdeen Proving Ground, MD 21005-5069

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-8281

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Compressive sensing is a signal processing technique that takes advantage of a signal's sparsity to reduce sampling requirements. This can be used to improve system resolution, frame rate, power consumption, and memory usage. Most compressive sensing research has focused on developing sensing systems and demonstrating their performance sensing preexisting objects. Little research has been directed toward designing the objects being sensed. This report addresses this overlooked area. Simple examples are shown demonstrating the advantage of modifying an object’s sparsity to increase or decrease compressive sensing performance. This leads to more complex object recognition examples where the object's sparsity must be balanced against its quality for optimal performance. In these cases, simulation results show that there are significant gains when one designs for compressive sensing.

15. SUBJECT TERMS
compressive sensing, fonts, optical character recognition, quick response codes, camouflage, compression

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
Unclassified	Unclassified	Unclassified	UU	30	Michael L Don
					19b. TELEPHONE NUMBER (Include area code)
					410-306-0775

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

Compressive sensing (CS) is a relatively new field that has caused a lot of excitement in the signal processing community. It has superseded Shannon's time-honored sampling theorem, which states that the sampling rate of a signal must be at least twice its highest frequency.[1] In CS, the necessary sampling rate depends on the sparsity of signal, not its highest frequency, reducing sampling requirements for many signals that exhibit natural sparsity.[2,3] This compression happens on the hardware level, allowing systems to be designed with benefits ranging from increased resolution and frame rates to decreased power consumption and memory usage.[4] Despite this enthusiasm for CS and the large quantity of research being performed, the number of commercial systems that use CS is relatively few. The problem of designing a CS strategy that increases functionality while actually reducing overall system cost has not been solved in many areas.[5] This is a developing field where not only are new applications for CS still being developed but also fundamental aspects of CS theory are still evolving.

Even though CS has not become ubiquitous at this early date, one can look forward to a time in which it plays an important role in many sensing systems. Considering this possible future, it is important not only to properly design the CS sensor, but to also consider how the objects being sensed can be designed to increase overall system performance. This idea is not unique to CS; examples of designing objects to improve the performance of specific technologies can be found in other areas as well. The image on the left of Fig. 1 shows a moire pattern caused by interference between the shirt's stripes and the pattern of the imaging array.[6] When television (TV) newscasters are told to avoid clothes that could cause these patterns,[7] the objects being sensed (the newscasters) are effectively being designed to increase the performance of the sensing system (the TV cameras). Another example is the magnetic ink character recognition (MICR) font shown in Fig. 2. This font is used on checks and was designed not only to be readable by humans but also to increase the character recognition performance of MICR readers.[8]

Fig. 1 Moire pattern example

Fig. 2 MICR font

Before exploring how objects can be designed for CS, a short review of CS theory is presented. Next, simple examples are shown demonstrating the advantage of modifying an object's sparsity to increase or decrease CS performance. This leads to more complex object recognition applications where an object's sparsity must be balanced against other factors. Increasing an object's sparsity improves CS performance, resulting in higher reconstruction quality and improved object recognition. But the very act of increasing sparsity distorts the object, which can impair recognition. Simulation results show that by balancing these competing factors an optimal design can be achieved.

2. Compressive Sensing Theory

In traditional sampling, the value of the signal is digitized at specific points in time and space. In CS, a sample is formed by taking the sum of the signal projected onto a random code. More formally, given a signal $\boldsymbol{f} \in \mathbb{R}^{\mathrm{N}}$ and a random code $\boldsymbol{h} \in \mathbb{R}^{\mathrm{N}}$, a compressive sample g is

$$g = \boldsymbol{h}^T \boldsymbol{f}. \tag{1}$$

Each compressive sample uses a different random code. For M measurements, these codes become the rows of a sensing matrix $\boldsymbol{H} \in \mathbb{R}^{M \times N}$ to produce the measurement vector $\boldsymbol{g} \in \mathbb{R}^{M}$ given by

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f}. \tag{2}$$

CS requires $\boldsymbol{f}$ to be sparse, but not necessarily in the standard basis; it can be sparse in any basis. Given the sparse coefficients $\boldsymbol{\theta}$ of $\boldsymbol{f}$ in basis $\boldsymbol{\Psi}$, the measurements can be rewritten as

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{\Psi}\boldsymbol{\theta} = \boldsymbol{A}\boldsymbol{\theta}, \tag{3}$$

where $\boldsymbol{f} = \boldsymbol{\Psi}\boldsymbol{\theta}$ and the system matrix $\boldsymbol{A}$ is defined as $\boldsymbol{A} = \boldsymbol{H}\boldsymbol{\Psi}$. Given that $N > M$, there are an infinite number of solutions to Eq. 3, but CS theory states that if there are a sufficient number of measurements, the sparsest solution will recover the original signal. This can be found through L1-norm minimization denoted as

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_1 \quad \text{subject to} \quad \boldsymbol{g} = \boldsymbol{A}\boldsymbol{\theta}. \tag{4}$$

The number of measurements required to reconstruct $\boldsymbol{f}$ is

$$M \geq C \cdot \mu^2 \cdot \|\boldsymbol{\theta}\|_0 \cdot \log N, \tag{5}$$

where C is a positive constant, $\|\boldsymbol{\theta}\|_0$ is the number of nonzero values in $\boldsymbol{\theta}$, and μ is a small constant determined by the structure of $\boldsymbol{A}$ called mutual coherence. The important point is that the number of measurements is linearly related to its sparsity $\|\boldsymbol{\theta}\|_0$, but only logarithmically related to the signal's size, allowing large sparse signals to be sampled with relatively few measurements.

Figure 3 shows a simple CS example. A sparse signal $\boldsymbol{f}$ of length $N = 64$ is shown at the top. There are 4 nonzero values (i.e., $\|\boldsymbol{\theta}\|_0 = 4$). The next plot shows the random code $\boldsymbol{h}$ used for the first measurement produced from a zero mean, unit variance Gaussian random variable. A compression ratio of 0.25 is used, giving a total of $M = N/4 = 16$ measurements. The random code for each measurement appears as a row in the sensing matrix $\boldsymbol{H} \in \mathbb{R}^{16\times 64}$. After the measurement vector $\boldsymbol{g}$ is determined by Eq. 2, L1-norm minimization is used to recover the original signal. The recovered signal is almost identical to the original, with the error shown in the bottom plot. Figure 4 shows the spectrum of this signal, assuming a sampling frequency of 1 Hz. Since the signal has frequency components up to 0.5 Hz, Shannon's sampling theory would predict that the sampling rate must be twice this frequency to recover the signal (i.e., all 64 samples are necessary to recover the original signal). In this example, CS was able to recover the signal using only 25% of this number of measurements.

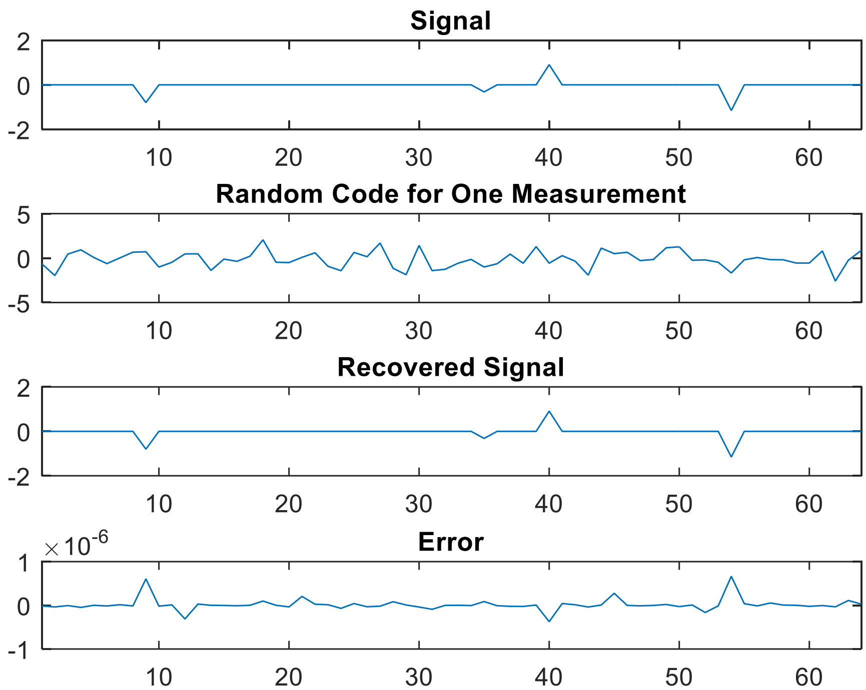

Fig. 3 **Simple CS example**

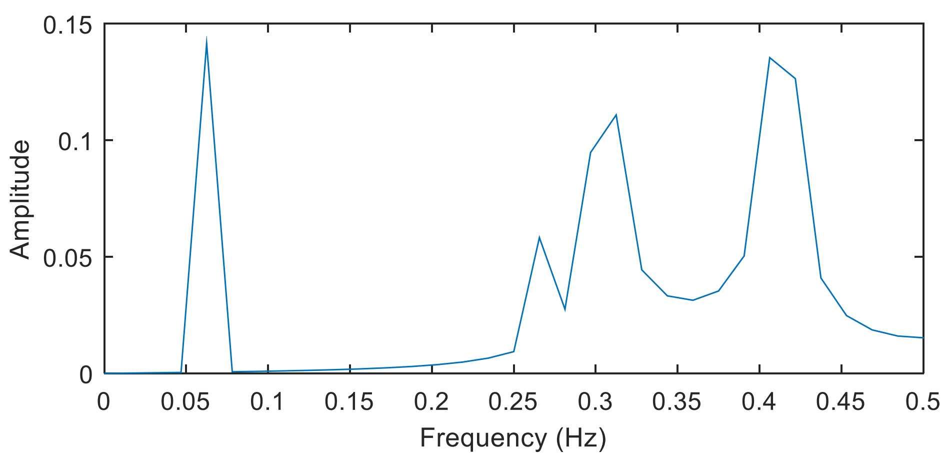

Fig. 4 **Single-sided amplitude spectrum of signal in Fig. 3**

In the previous example, the signal was sparse in the standard basis and therefore it was not necessary to change the basis during CS recovery. Figure 5 shows an example where the signal is non-sparse in the standard basis, but sparse in the discrete cosine basis $\boldsymbol{\Psi}$. The top plot shows the signal from the previous example, now used as sparse discrete cosine transform (DCT) coefficients $\boldsymbol{\theta}$. The next plot shows the non-sparse signal $\boldsymbol{f}$ in the standard basis, after applying an inverse DCT on the DCT coefficients. L1-norm minimization is used to recover the DCT, which can then be transformed into the standard basis. Once again the reconstruction is almost identical, with a small error shown on the bottom plot. The rest of this report uses, unless otherwise noted, a binary block diagonal sensing matrix,[9] a wavelet basis, and the gradient projection for sparse reconstruction[10] recovery algorithm.

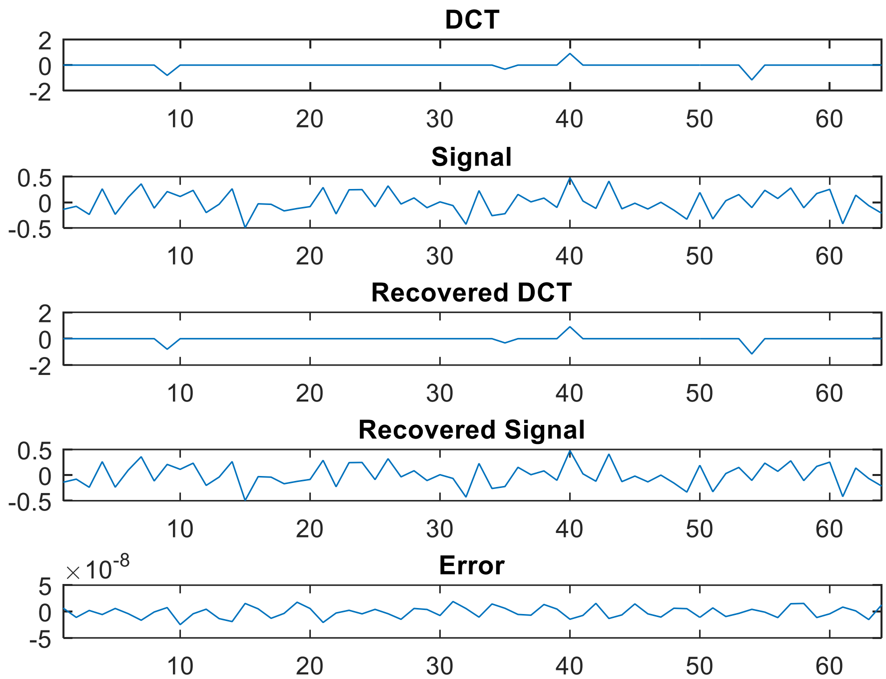

Fig. 5 CS example using the DCT as the sparse basis

Although CS is a powerful sensing technique, it does have disadvantages compared to traditional sampling. The L1-norm minimization algorithms used for recovery are computationally intensive and challenging to implement in real-time systems, whereas in traditional sampling the data is immediately available. In addition, the cost and complexity of the hardware necessary to produce the compressive samples frequently outweighs the benefits of CS. Many researchers are optimistic that these problems will be solved, allowing CS to become widespread in many sensor systems. Currently, however, there are relatively few examples of commercially successful CS hardware. This presents another benefit of designing for CS. By lowering expectations and concentrating on environments that have been designed to provide high CS performance, practical applications may be found for CS that had once been thought to be unrealistic.

In order to illustrate an example of CS, see its advantages and disadvantages, and gain a broader understanding of the benefits of designing for CS, we look at an early implementation of a CS imager, Rice University's single-pixel camera.[11] A system block diagram is shown in Fig. 6. The scene is focused on a digital micromirror device (DMD), which has a matrix of small mirrors that can be individually programmed to point toward or away from the detector. This is used to impose a pseudorandom block-unblock mask on the scene, which is represented by the binary code in the figure. All of the light directed toward the single detector is summed and digitized to create a single CS sample. The random pattern of the DMD is changed for each measurement to create a number of compressive samples. The diagram shows the CS samples being wirelessly transmitted before the image is recovered.

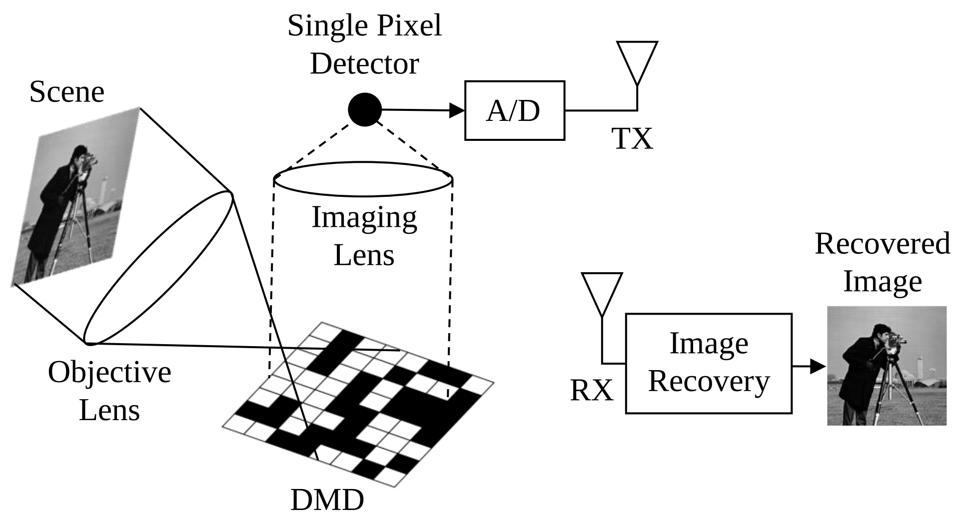

Fig. 6 Single-pixel camera diagram

This CS camera has a number of advantages:

- The resolution of the imager is increased from a single pixel to the number of mirrors of the DMD, which is available at resolutions up to 3840 × 2160.[12]
- There are fewer measurements taken, which can reduce power consumption and increase the frame rate when compared to raster scanning.
- The samples are already compressed when acquired, reducing memory and bandwidth requirements.
- For each sample, the single detector receives more light than a high-resolution detector, which can reduce noise in some applications. Single-pixel sensors do not experience the pixel-bleeding problems of larger infrared (IR) imaging arrays and can therefore use inexpensive cooling systems for IR applications.[13]

Although these advantages sound impressive, this CS imager might not be practical for many applications. The benefits of CS must be carefully weighed against the following disadvantages:

- The DMD itself is an expensive instrument that increases the cost and complexity of the system.
- Computationally intensive algorithms are needed to recover the image.
- Although CS increases the frame rate when compared to other scanning technologies, it is still much slower than single-snapshot imaging arrays.
- Like other scanning technologies, the slow acquisition rate makes it unsuitable for many mobile applications.
- The DMD can only be used for wavelengths from 300 to 2700 nm.[14] This excludes some of the most costly sensors that would have benefited the most from CS.

By examining these advantages and disadvantages, a viable design for this single-pixel camera begins to emerge. This imager is well suited for IR applications where a single-pixel sensor can provide significant cost savings over high-cost IR imaging arrays. Ideally, the target application should be static, have low frame rate requirements, and only require the images for postprocessing. Low-light environments can take advantage of the increased light seen by the detector. Underlying many of these considerations is the basic CS performance, evaluated by the number of measurements needed to obtain the desired image quality. Greater CS performance can translate into higher-resolution images, faster frame rates, and reduced power, making it an essential parameter for practical CS hardware design. A camera design whose disadvantages outweigh its advantages in general environments may become more practical if its CS performance is improved by limiting it to sparse environments. This idea is already applied by many researchers who use artificially sparse data sets to enhance the performance of their compressive sensors. Technical papers have been known to use simple shapes,[15] Lego men,[16] or—in the case of Rice's CS camera—a single letter to demonstrate CS performance.

Figure 7 shows an example of the dramatic difference in CS performance between using a carefully selected sparse image and a typical image. On the left side of Fig. 7 is the classic cameraman image and on the right side is an "R" similar to an example image used by the single-pixel camera. The top plot of Fig. 8 shows the first 10,000 wavelet coefficients of these images in descending order. The sharper decline of the "R" coefficients compared to the cameraman coefficients indicate the greater sparsity of the "R" image. This translates to the greater CS performance shown in the lower plot, with the "R" image reconstruction consistently exhibiting a higher peak signal-to-noise ratio (PSNR) than the cameraman image over a range of compression ratios. As in the original single-pixel camera article, a total variation reconstruction algorithm was used here that is similar to L1 minimization.[17]

Fig. 7 Two example images, the cameraman and the letter "R"

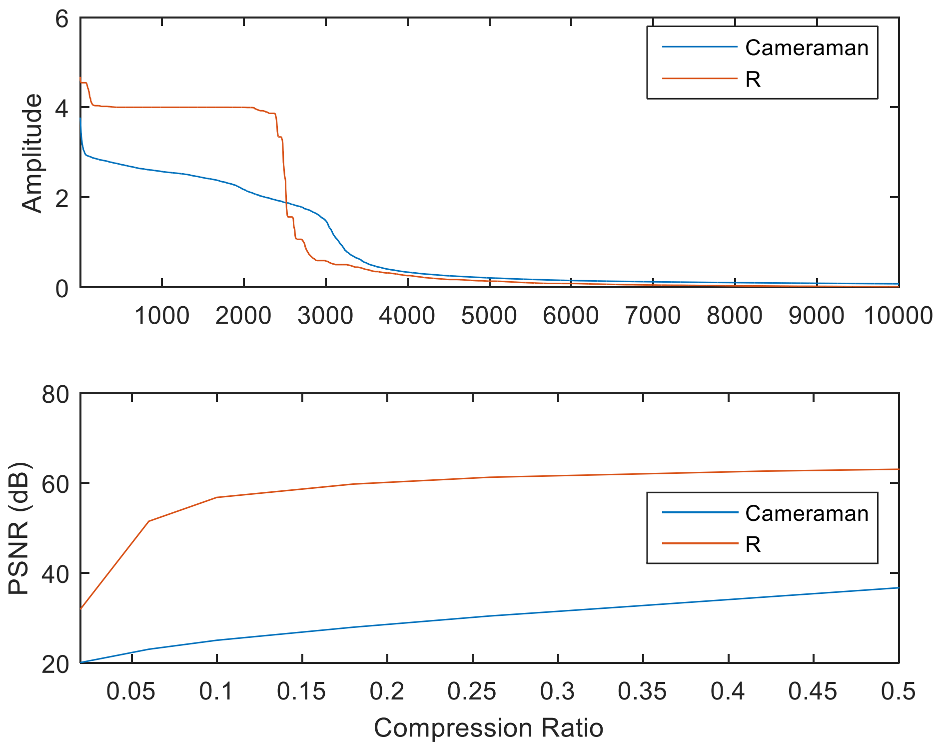

Fig. 8 The first 10,000 wavelet coefficients in descending order (top) and the CS reconstructed PSNR vs. compression ratio (bottom) of the images in Fig. 7

Researchers use sparse example images to simplify problems and advance their research. Instead of just using sparse data as a stepping stone to more challenging environments, this report considers designing for CS an area of research in its own right.

3. Designing for Sparsity

In the introduction, we used newscasters avoiding clothing that can cause moire patterns as an example of designing the objects being sensed to increase the performance of the sensing system. In a similar vein, given that CS performance is proportional to sparsity, it follows that scenes can be designed with greater sparsity to increase the performance of CS cameras. Let us look at an example.

The 44th President of the United States Barack Obama[18] is addressing the nation framed by artwork in the background. Since abstract expressionism was the first truly American art that transformed the US into the center of the art world,[19] an abstract expressionist painting is chosen for this important event. If the video camera uses CS with a compression ratio of 0.35, should a Jackson Pollock or Mark Rothko painting be chosen? The upper 2 images in Fig. 9 show the original uncompressed and CS-recovered images using the Jackson Pollock painting "Number 1" for the background.[20] Figure 10 shows a comparison of the Pollock image's wavelet coefficients, vectorized and sorted in descending order, with those of the Rothko image. Observe that Jackson Pollock paintings are decidedly not sparse, containing many more significant wavelet coefficients than the Mark Rothko painting. This results in a disappointing PSNR of 20 dB in the recovered Pollock image in the rectangle containing the president. The lower 2 images using Mark Rothko's much sparser "Orange and Yellow" canvas[21] give a more acceptable PSNR of 29 dB for the president's face. Clearly, in this scenario, designing the environment for sparsity greatly enhanced image quality.

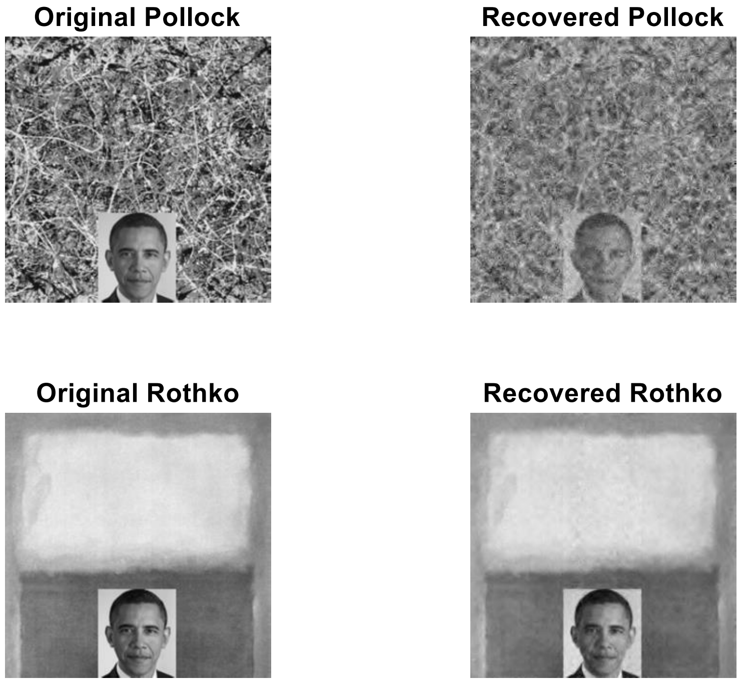

Fig. 9 **Examples of CS imaging in non-sparse (top) and sparse (bottom) environments**

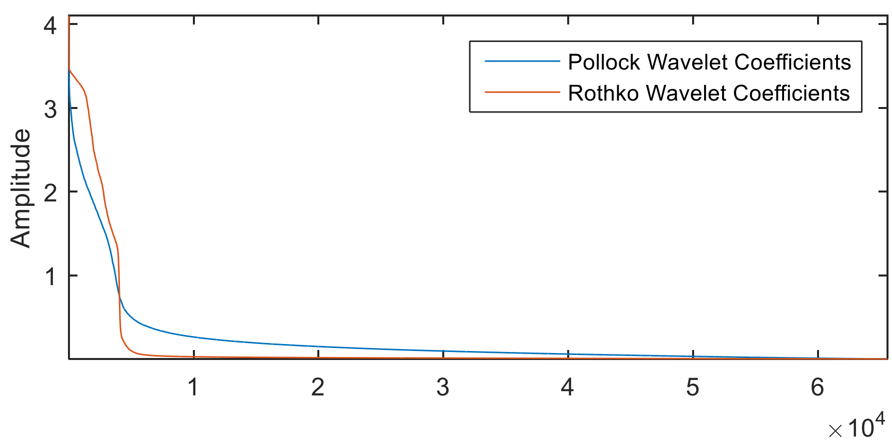

Fig. 10 **Comparison of the wavelet coefficients of the 2 images in Fig. 9**

This simple example is not unique to compressive sensing; any normal image compression produces similar results. Image compression can be accomplished by retaining the amplitude and location of only the largest coefficients in a sparse basis. Figure 11 shows an example of compressing the images from Fig. 9 by retaining the largest 10% of the wavelet coefficients. Since the Pollock image is not sparse, this process removes a significant amount of information, creating the blurred image in the upper right. The Rothko image is sparse, therefore little information is lost during the compression process resulting in the high-quality image in the lower right. Although the examples here and in the next section also apply to normal compression, we discuss applications in Section 5 that are unique to CS.

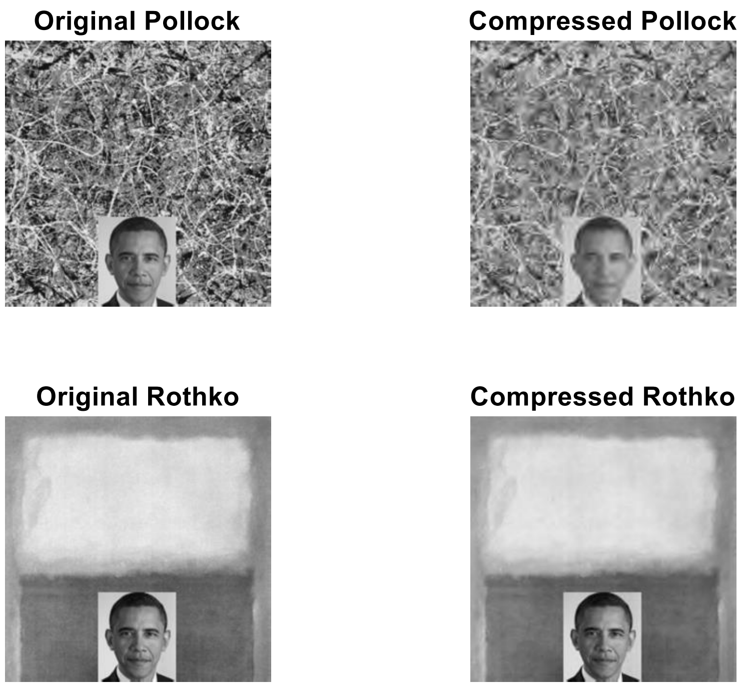

Fig. 11 Example of normal image compression for a non-sparse (top) and sparse (bottom) image

4. Designing for Non-Sparsity

In general, CS applications seek to increase image quality, but applications can be envisioned where we desire to decrease CS performance. Imagine a futuristic ninja battling CS-equipped unmanned aerial vehicles. What clothing would be more effective in confusing the CS sensors of these drones, the black outfit of the traditional dramatized ninja[22] or a CS camouflage consisting of a random noise pattern? Using a compression ratio of 0.35, the black outfit in the top of Fig. 12 does not provide much of a disguise, resulting in a PSNR of 35 dB for the rectangle containing the ninja's eyes. In the bottom images, the non-sparsity of the noise acts as a CS camouflage, hiding the eyes with a PSNR of 16 dB. Figure 13 shows a comparison of the black ninja's wavelet coefficients, vectorized and sorted in descending order, with those of the CS camouflage image showing the dramatic decrease in sparsity possible when one designs for non-sparsity.

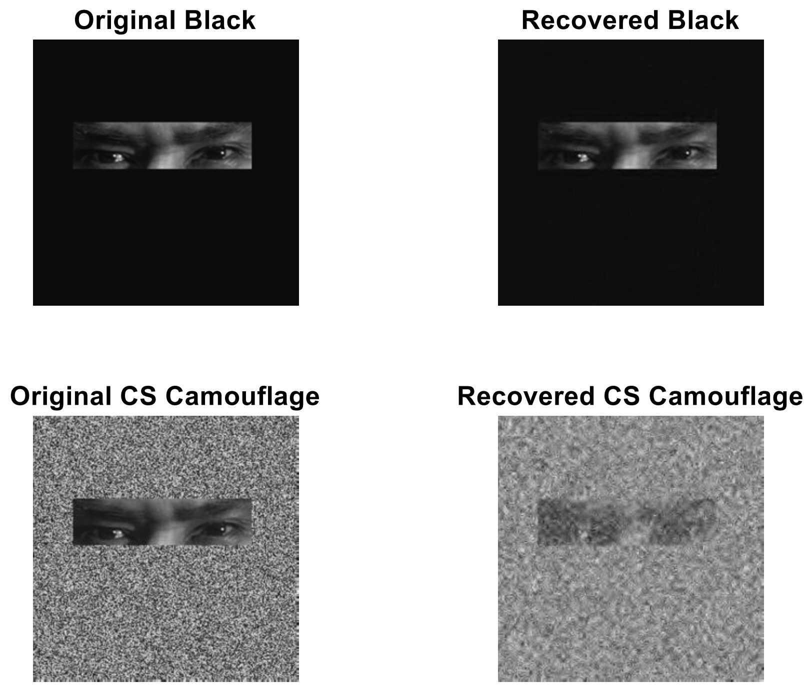

Fig. 12 Compressive camouflage example

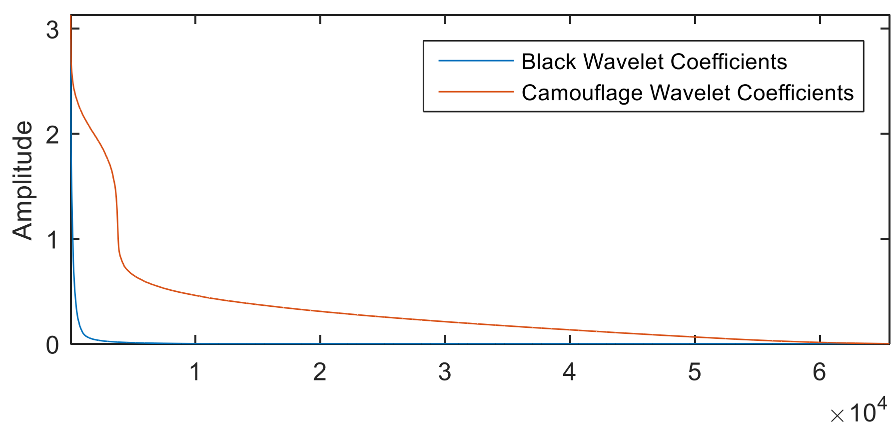

Fig. 13 Comparison of the wavelet coefficients of the 2 images in Fig. 12

5. Designing for Recognition

In the previous examples the relationship between sparsity and CS performance was straightforward: the sparser the image, the greater the CS performance. In Section 3, this relationship was used to increase the quality of the recovered image, while in Section 4, it was purposely used to obscure the image. Some applications exist, however, where image sparsity must be balanced against other factors. One example is using a compressive imager to perform optical character recognition

(OCR). Sparse compressive fonts can be designed to improve performance when imaged by a CS camera. If we make the fonts too sparse, however, it might degrade the quality of the fonts to the extent that they will be unrecognizable. Does it make sense to try to design fonts for CS imaging to improve OCR? Intuition suggests yes, since typically the sparsity of an image can be decreased considerably without significantly affecting image quality, while problems with CS recovery due to non-sparsity can create artifacts in the image that may hamper OCR.

Figure 14 shows example text in an unmodified Ariel Black bold font in the upper left. A CS-recovered version using the DCT and a compression ratio of 0.2 is shown on the lower left, with thresholding applied in the lower right. This is the maximum compression at which an OCR algorithm[23] was able to correctly identify the text in the thresholded image. Figure 15 shows a modified font in the upper right created using only the largest 3% of DCT coefficients. Here the text was able to be successfully identified when only using a compression ratio of 0.15. The fact that the OCR algorithm failed to recognize the original font using a compression ratio of 0.15, but successfully recognized the modified font, shows that reducing the font's sparsity increased OCR performance.

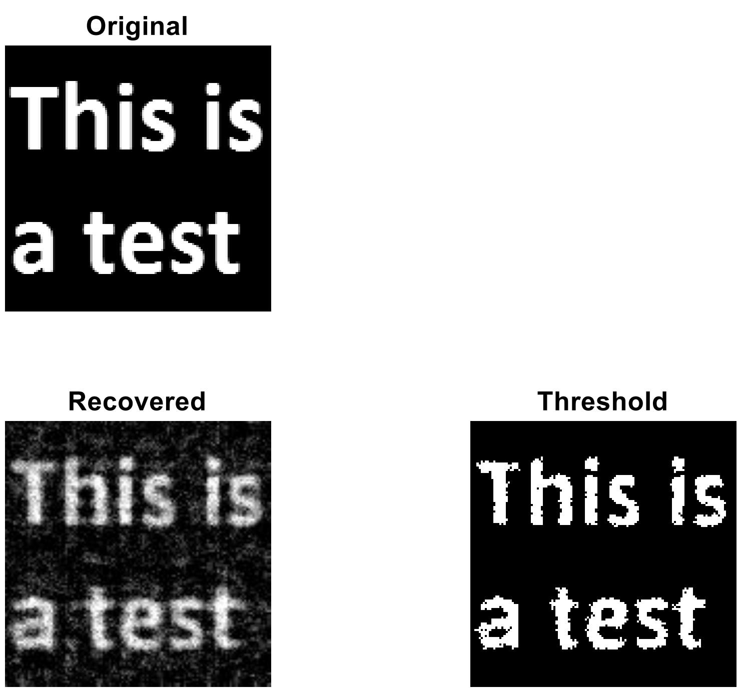

Fig. 14 CS OCR example using a standard font

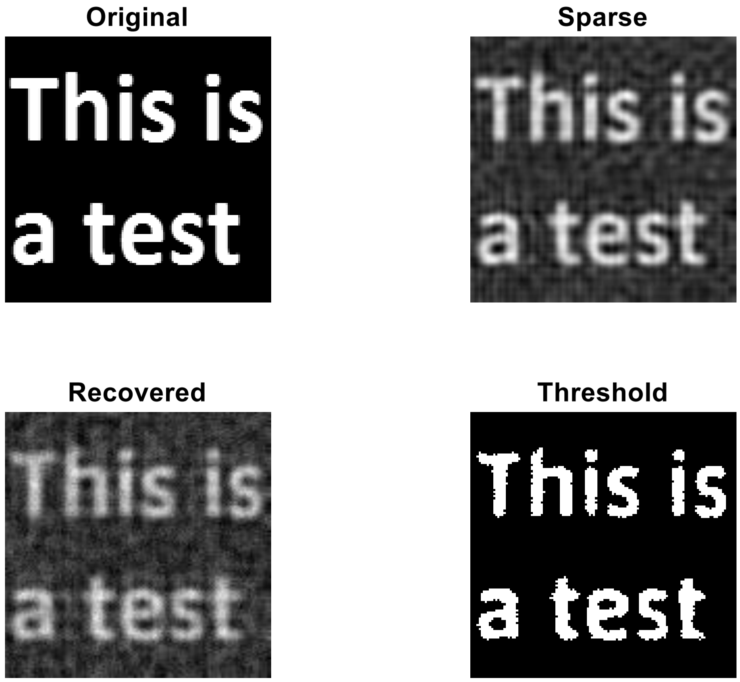

Fig. 15 CS OCR example using a compressive font

In order to analyze the relationship between the font's sparsity and its CS performance, the example in Fig. 15 was repeated over a range of sparsity values and compression ratios. In this context, sparsity refers to the percent of DCT coefficients from the standard text image used to create the sparse image. Figure 16 shows the resulting PSNR of the recovered text images for this experiment. Figure 17 indicates successful character recognition in yellow. As expected, there is a balance between sparsity and image quality that yields optimal results. Moreover, this balance in not identical for all compression ratios. A sparsity of 3% allows correct text identification using only a compression ratio of 0.15. A sparsity of 2.5% works for a compression ratio of 0.175, but not for 0.15. Since image sparsity should improve recovery, it is safe to assume that in this case the degradation of the image due to the sparsity is combining with the effects of CS recovery to prevent correct character identification. Thus the sparsity, CS reconstruction error, and image quality all have to be balanced in order to achieve optimal performance.

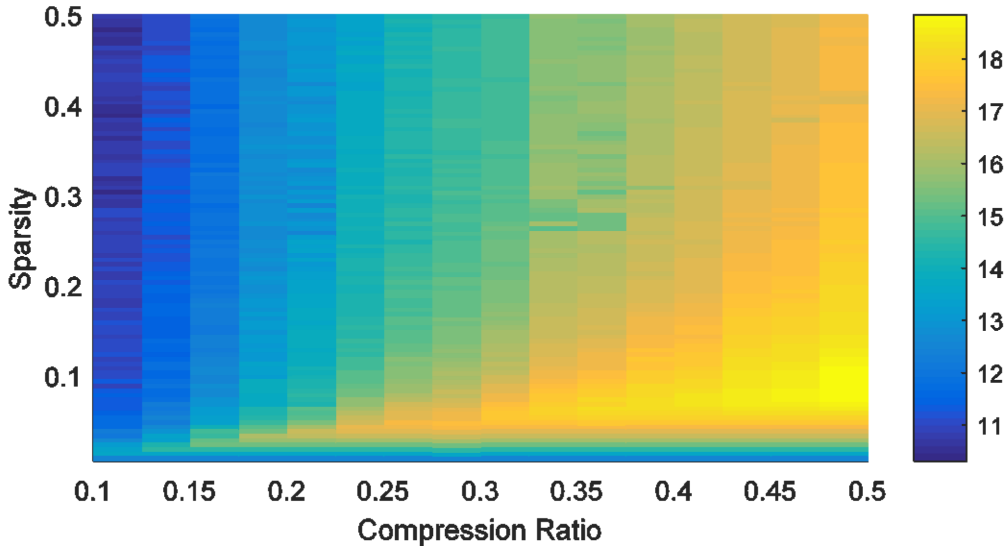

Fig. 16 PSNR of the recovered OCR image over a range of sparsity and compression ratios

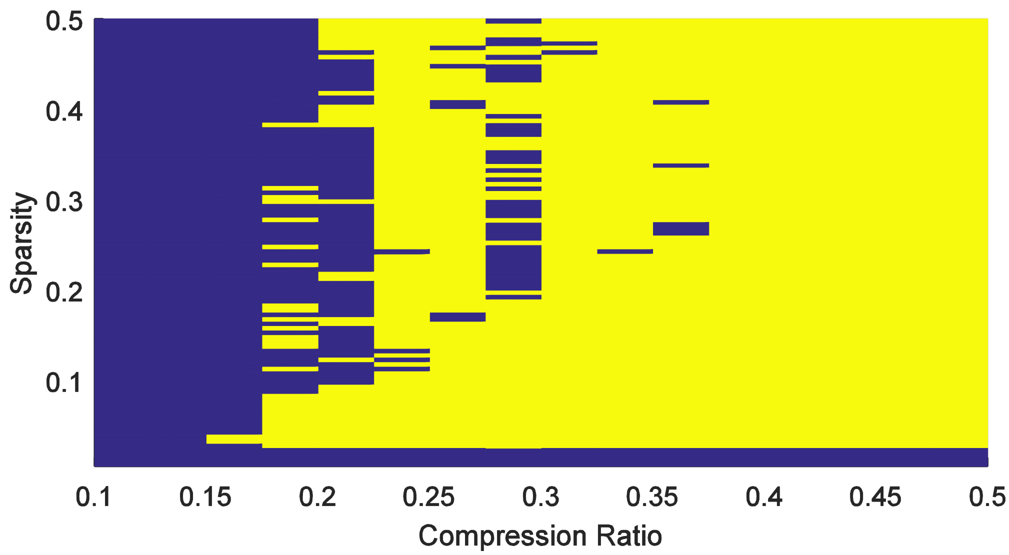

Fig. 17 OCR results from simulations in Fig. 16. Yellow indicates the recovered text was correctly recognized.

In Section 3, we noted that designing for sparsity applies to normal image compression as well as CS. The same is true for the CS camouflage example in Section 4. However, the examples in this section of designing for recognition do not apply to normal image compression. The final image will appear the same whether the text is imaged and then compressed, or if it is first compressed and then imaged. Increasing the sparsity of the physical text serves no purpose. If the text image is compressed, the only factor that must be considered is the maximum amount of compression before OCR errors begin to occur. The upper right of Fig. 18 shows the text compressed by retaining 0.9% of the DCT coefficients. This was the maximum compression before OCR errors occurred. This is a less interesting scenario than designing fonts for CS where both the font's sparsity and CS recovery error must be taken into account to optimize font design.

Fig. 18 An OCR example using normal image compression

An experiment similar to the compressive font experiment was conducted on QR codes with similar results. Figure 19 shows an original, non-sparse image of a QR code in the upper left encoding the text, "This is a test".[24] A CS compression ratio of 0.44 allowed for correct decoding of the message in the recovered image shown in the upper right. Figure 20 shows a QR code using only the largest 5% of wavelet coefficients. Here the compression ratio threshold for successful recognition was 0.1. Once again, increasing the image's sparsity increases recognition performance.

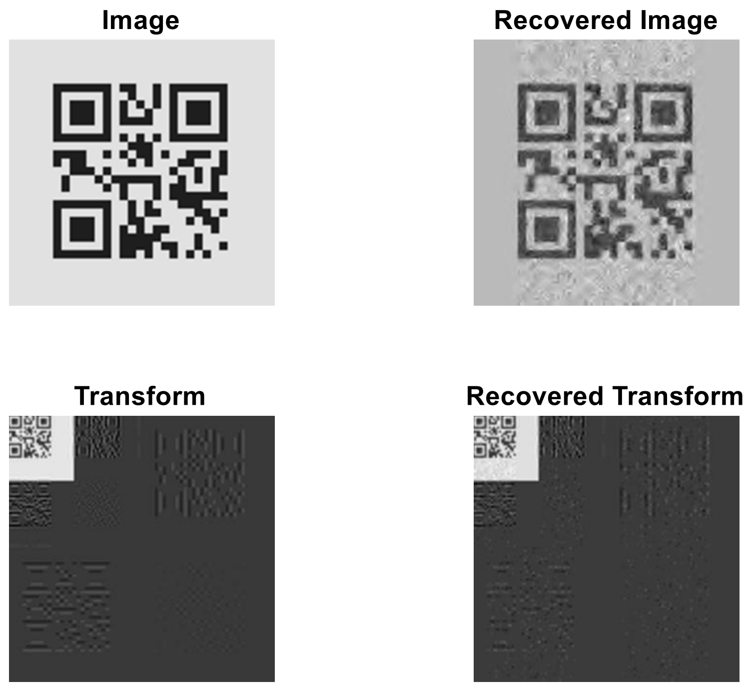

Fig. 19 CS non-sparse QR example

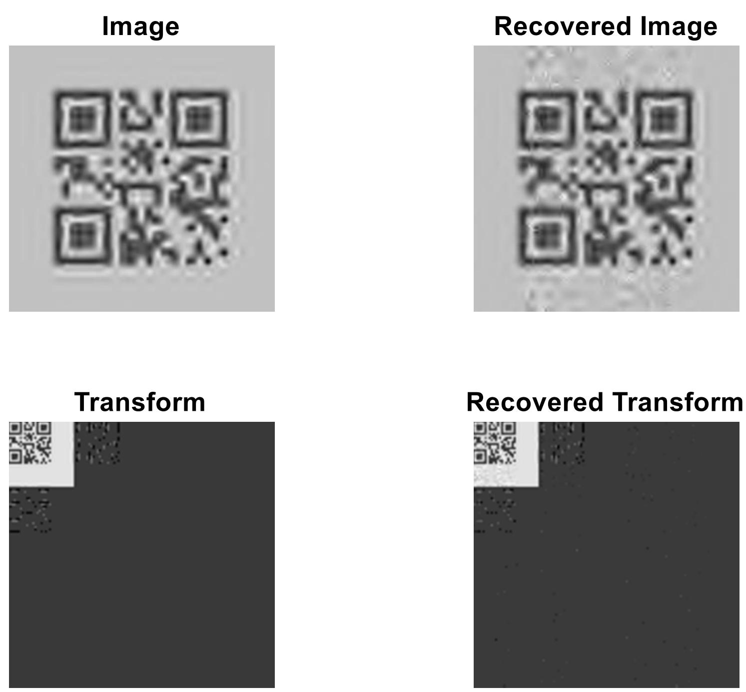

Fig. 20 CS sparse QR example

Figure 21 shows the resulting PSNR of the recovered QR codes over a range of sparsity values and compression ratios. Figure 22 indicates the successful decoding of the simulations from Fig. 21 in yellow. Here a sparsity of 5% resulted in the best CS performance, allowing correct decoding using a compression ratio of 0.1. Comparing Fig. 22 to the analogous OCR results in Fig. 17, the CS performance deteriorates much more quickly for QR codes than for OCR as the images become less sparse. This is probably due to the QR code design that has many small features and hence is naturally less sparse than the OCR example, as shown in Fig. 23.

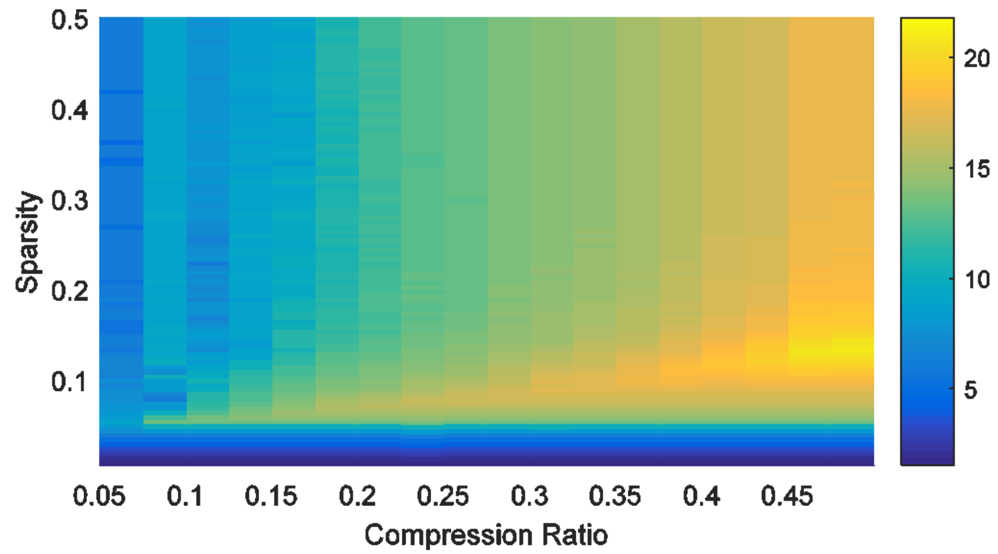

Fig. 21 PSNR of the recovered QR code over a range of sparsity and compression ratios

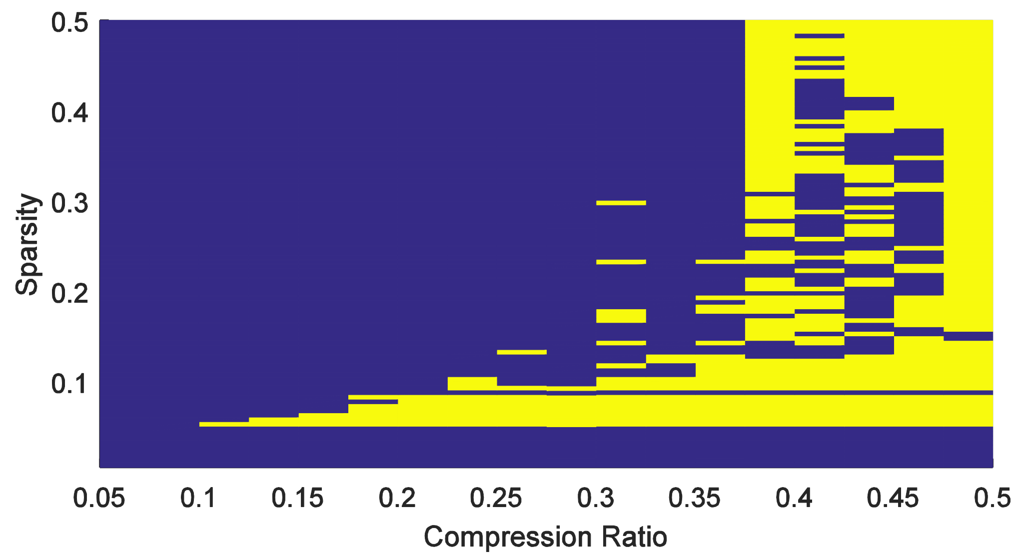

Fig. 22 QR code decoding results from simulations in Fig. 21. Yellow indicates the recovered text was correctly decoded.

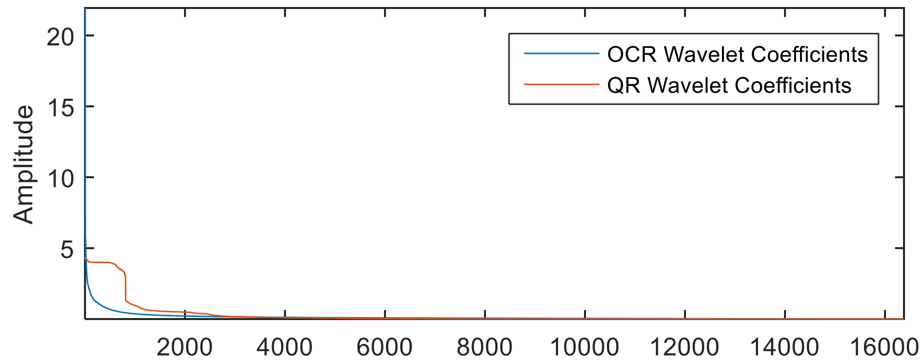

Fig. 23 Comparison of the wavelet coefficients of the OCR and QR code example images

6. Conclusion

This report has explored the concept of designing for CS. Although at this time CS has not become commercially widespread, many of the sensing systems of the future may incorporate CS into their design. If so, there will be instances where it will be advantageous to design the objects being sensed for CS in addition to designing the sensing systems themselves. Simple examples were shown demonstrating the advantage of modifying an object's sparsity to increase or decrease CS performance. This led to more complex object recognition examples where the object's sparsity was balanced against its quality to achieve optimal performance. In these cases, simulation results showed that there were significant gains when one designs for CS.

7. References

1. Shannon CE. Communication in the presence of noise. Proc IRE. 1949;37(1):10–21.
2. Candès EJ, Romberg JK, Tao T. Stable signal recovery from incomplete and inaccurate measurements. Commun Pure Appl Math. 2006;59(8):1207–1223.
3. Donoho DL. Compressed sensing. IEEE Trans Inf Theory. 2006;52(4):1289–1306.
4. Candès EJ, Wakin MB. An introduction to compressive sampling. IEEE Signal Process Mag. 2008;25(2):21–30.
5. Strohmer T. Measure what should be measured: progress and challenges in compressive sensing. IEEE Signal Process Lett. 2012;19(12):887–893.
6. Wikimedia Commons. File:Shirt with moiré caused by aliasing.jpg. 2015 Mar 23 [accessed 2017 Oct 19]. https://commons.wikimedia.org/wiki/File:Shirt_with_moir%C3%A9_caused_by_aliasing.jpg.
7. PLSN. A word About Moiré [accessed 2017 Oct 19]. http://plsn.com/current-issue/35-video-world/12215-a-word-about-moire.html.
8. Wikipedia. Magnetic ink character recognition. 2017 Dec 1 [accessed 2017 Oct 19]. https://en.wikipedia.org/wiki/Magnetic_ink_character_recognition.
9. Do TT, Tran TD, Gan L. Fast compressive sampling with structurally random matrices. Proceedings of 2008 IEEE International Conference on Acoustics, Speech and Signal Processing; 2008 Mar 31–Apr 4; Las Vegas, NV. IEEE; 2008.
10. Nowak RD, Wright SJ, Figueiredo MA. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. IEEE J Sel Top Signal Process. 2007;1(4):586–597.
11. Duarte MF, Davenport MA, Takhar D, Laska JN, Sun T, Kelly KE, Baraniuk RG. Single-pixel imaging via compressive sampling. IEEE Signal Process Mag. 2008;25(2):83–91.
12. Texas instruments DLP660TE chipset [accessed 2017 Oct 20]. https://www.ti.com/product/DLP660TE.
13. Team single pixel camera. c2007 [accessed 2017 Oct 20]. http://elec424.rice.edu/spc/.

14. Zhang K, Huang Y, Yan J, Sun L. Dynamic infrared scene simulation using grayscale modulation of digital micro-mirror device. Chin J Aeronaut. 2013 30 Apr;26(2):394–400.

15. Ghosh A, Powers MA, Patel VM. Computational LADAR imaging. Appl Opt. 2017 Jan 20;56(3);B191–7.

16. Don ML, Fu C, Arce GR. Compressive imaging via a rotating coded aperture. Appl Opt. 2017 Jan 20;56(3);B142–53.

17. Candès EJ, Romberg JK. l1-magic: recovery of sparse signals via Convex programming [accessed 2018 Jan 12]. http://statweb.stanford.edu/~candes/l1magic/downloads/l1magic.pdf.

18. Wikimedia Commons. File:Obama portrait crop.jpg. 2017 Jan 15 [accessed 2017 Oct 19]. https://commons.wikimedia.org/wiki/File:Obama_portrait_crop.jpg.

19. Janson HW, Janson AF. History of art. 3rd ed. New York (NY): HN Abrams; 1986. p. 740.

20. Number 1, 1949. Collection, Jackson Pollock. Los Angeles (CA): The Museum of Contemporary Art [accessed 2017 Oct 19]. https://www.moca.org/collection/work/number-1.

21. Orange and Yellow, 1956. Collection, Mark Rothko. Buffalo (NY): Albright-Knox Art Gallery [accessed 2017 Oct 19]. https://www.albrightknox.org/artworks/k19568-orange-and-yellow.

22. Turnbull S. Ninja AD 1460–1650. Oxford (UK): Osprey; 2004.

23. Matlabcentral. Optical character recognition (lower case and space included) by Ankit Saroch [accessed 2017 Oct 19]. https://www.mathworks.com/matlabcentral/fileexchange/31322-optical-character-recognition-lower-case-and-space-included.

24. QR code encode and decode by Lior Shapira [accessed 2017 Oct 19]. https://www.mathworks.com/matlabcentral/fileexchange/29239-qr-code-encode-and-decode.

List of Acronyms

ARL	US Army Research Laboratory
CS	compressive sensing
DCT	discrete cosine transform
DMD	digital micromirror device
IR	infrared
OCR	optical character recognition
PSNR	peak signal-to-noise ratio
QR	quick response
TV	television

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIR ARL IMAL HRA RECORDS MGMT RDRL DCL TECH LIB
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
26 (PDF)	ARL RDRL WML E F FRESCONI RDRL WML F B ALLIK B ACKER T BROWN S BUGGS E BUKOWSKI J COLLINS J CONDON B DAVIS M DON D EVERSON D GRZYBOWSKI R HALL J HALLAMEYER M HAMAOUI T HARKINS M ILG B KLINE J MALEY C MILLER P MULLER B NELSON D PETRICK K PUGH N SCHOMER B TOPPER

INTENTIONALLY LEFT BLANK.